Weight Exchange in Decentralized Distributed Machine Learning for Resource-Constrained IoT Edges

Naoya Yokota[1,a)] Yuko Hara-Azumi[1,b)]

Abstract: Although a Gossip Stochastic Gradient Descent (SGD) algorithm is known to be suitable for decentralized distributed machine learning, it has a non-convergence problem for heterogeneous datasets between multiple devices. In this paper, we propose a Gossip Swap SGD to address this problem by employing a weight swapping method between devices. Our evaluation demonstrated that our proposed method successfully improves higher accuracy without increasing computation load than the original Gossip SGD.

Keywords: Gossip Stochastic Gradient Descent, Noncentralized Distributed Machine Learning, Weight Swapping

1. Introduction

Nowadays, an increase in number of IoT devices is rapidly growing the amount of data collected at those devices. In order to utilize such data, edge AI technology is expected to be deployed. Since data aggregation in one place (e.g., server) for learning cannot be always feasible, for instance due to a heavy communication load, distributed machine learning (ML) on the edge devices is attracting attentions [1]. Federated Learning [4] is one of promising techniques for this purpose, but is applicable to a limited topology where a centralized device (e.g., edge server) collects and updates the learning parameters of edge devices communicating with it. This assumption would lead to vulnerability to network failure. Another approach is Gossip SGD targeting decentralized distributed ML [3] that does not require a centralized device to let devices learn, mainly in environments that devices have data for same classes (i.e., homogeneously-distributed dataset).

Unlike the aforementioned existing works, our proposed method in this paper, named *Gossip Swap SGD*, has a contribution of resolving a non-convergence problem that the existing Gossip SGD methods face in heterogeneously dataset environments. Our method employs weights exchanges between connected devices, leading to a better convergence in weight update and hence better accuracy. Through evaluation, we demonstrate that our proposed method outperforms the original Gossip SGD in two different Wireless Sensor Network topologies.

2. Gossip Stochastic Gradient Descent

The Gossip SGD is widely used for training ML in decentralized distributed networks (e.g., WSN), where each device gathers and stores the data on itself. Alg. 1 describes the pseudo code of Gossip SGD running on a device i in WSN. First, the weight w_i^0, the weight w_i of the neural network on the device i at iteration 0, is initialized (line 1). Then a mini batch x is generated from data set X held by the device i to run batch learning with optimizer SGD (lines 2-5), followed by weight update (line 4), which uses the learning rate α and derivative of the loss functionΔf. If the update count k reaches the upper bound (denoted as interval), another device j ($i \neq j$) is randomly selected from N_i which are devices connected to the device i to update w_i^t and w_j^t, the weights of device i and j at iteration t, by calculating the average of w_i^t and w_j^t (lines 7-10).

Algorithm 1 Gossip SGD on device i (input: data X)

```
 1: Initialization of $w_i^0$, $k = 0$
 2: for mini batch $x \in X$ do
 3:     $update\_variables(x, k, w_i^t)$
 4:     $w_i^{t+1} \leftarrow w_i^t - \alpha\Delta f(w_i^t, x)$
 5:     $k \leftarrow k + 1$
 6:     $communicate(N_i)$
 7:     if $k ==$ interval then
 8:         $j \leftarrow select\_device(\mathbb{N})$
 9:         $w_i^{t+1}, w_j^{t+1} \leftarrow average(w_i^t, w_j^t)$
10:         $k = 0$
11:     end if
12: end for
```

Since Gossip SGD assumes that identical dataset is available on all devices in the network (i.e., homogeneously distributed), it is confronted with a non-convergence problem in networks where data is *heterogeneously* distributed between devices.

3. Gossip Swap SGD

This work addresses the non-convergence problem in Gossip SGD to make it applicable to decentralized distributed networks where distinct datasets can be available on devices (hereafter we refer to such networks as our target environment) – this scenario is more realistic. In order to improve the weight update method in Gossip SGD, we focus on “weight swapping” that was originally proposed in Federated Learning for *centralized* distributed ML [2]. Since *averging* the weights in Alg. 1 could be a cause of the non-convergence problem in our target environment, we present a first attempt of employing weight swapping in Gossip

[1] Tokyo Institute of Technology, Meguro, Tokyo 152–8552, Japan
[a)] yokota.n.ad@m.titech.ac.jp
[b)] hara@cad.ict.e.titech.ac.jp

Algorithm 2 Gossip Swap SGD on device i (input: data X)

```
 1: Initialization of $w_i^{(0)}$, $k = 0$, $s = 0$
 2: for mini batch $x \in X$ do
 3:     $update\_variables(x, k, w_i^t)$
 4:     $w_i^{t+1} \leftarrow w_i^t - \alpha\Delta f(w_i^t, x)$
 5:     $k \leftarrow k + 1$
 6:     $communicate(N_i)$
 7:     if $k ==$ interval then
 8:         $j \leftarrow select\_device(N_i)$
 9:         if $s ==$ swap-interval then
10:             $w_i^{t+1} \leftarrow w_j^t$, $w_j^{t+1} \leftarrow w_i^t$
11:             $s = 0$
12:         else
13:             $w_i^{t+1}, w_j^{t+1} \leftarrow average(w_i^t, w_j^t)$
14:             $k = 0$, $s \leftarrow s + 1$
15:         end if
16:     end if
17: end for
```

SGD for *decentralized* distributed ML. This method aims to realize learning in a pseudo-homogeneous data environment by exchanging the weights that were updated for various data stored on multiple devices.

Alg. 2 describes the pseudo code of Gossip Swap. The procedures on lines 1-6 are the same as Alg. 1. Then, in our proposed method, when the update count k reaches the upper bound (line 7), another device j ($i \neq j$) is randomly selected from N_i which are devices connected to the device i, to simply swap w_i^t and w_j^t, the weights of device i and j at iteration t , if the swap-count s is equal to the swap-interval (lines 9-11). Otherwise, the weights w_i^t and w_j^t are averaged similarly as done in Alg. 1 (lines 12-14).

4. Evaluation

We simulated the original Gossip SGD and our proposed Gossip Swap SGD to evaluate the effectiveness of our proposed method. The device environment was a MacBook air 2017 model that has a CPU of Intel core i5 and 8GB memory. The neural network structure that we assumed to deploy on edge devices (e.g., a Raspberry Pi) consists of two convolutional layers, one pooling layer, and two fully-connected layers. Mini batch and epoch sizes were set to 200 and 14, respectively, for the MNIST dataset. As case studies, we evaluated the Gossip SGD and our Gossip Swap SGD on two WSN topologies illustrated in Fig. 1, where each device holds only subsets of the MNIST dataset and learns its weight based on it (e.g., Device 1 Topology 1 can use handwritten figures of 2, 3, and 4 for learning). Note that these target environments reflects the realistic situation where neighboring devices may hold partially same data and communicate with each other). In both of these two typologies, the interval k (in Algs. 1 and 2) and the swap-interval s (in Alg. 2) were both set to 1.

Due to space limitation, we disclose the results on accuracy of selected devices in Fig. 2, where the x-axis represents epochs and the y-axis shows accuracy for each class in the MNIST. For Device 1 in Topology 1, we compare accuracy of the Gossip SGD and our Gossip Swap SGD in Figs. 2(a) and 2(b), respectively. Similarly, we also compare accuracy of the Gossip SGD and our Gossip Swap SGD in Figs. 2(c) and 2(d), respectively, for Device 2 in Topology 2. As can be clearly seen from Figs. 2(a) and 2(c), the Gossip SGD cannot well converge learning for the dataset that was not held on the device due to the aforementioned non-convergence problem. On the other hand, our Gossip Swap SGD in Figs. 2(b) and 2(d) successfully achieved good accuracy for all classes. Similar tendencies were also observed for other devices in both topologies. These results well demonstrated that our Gossip Swap SGD can outperform the original Gossip SGD in realistic environments where data is heterogeneously distributed among multiple devices.

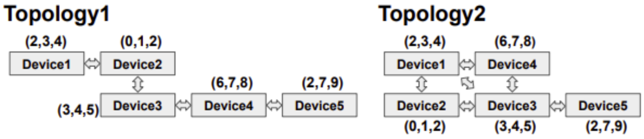

Fig. 1: Target environments

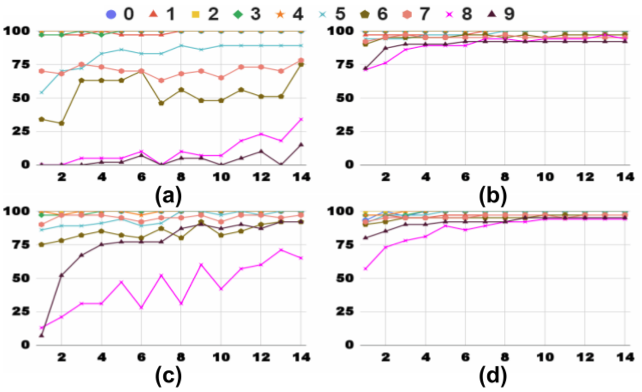

Fig. 2: Results: (a) Gossip SGD for Device 1 in Topology 1, (b) Gossip Swap SGD for Device 1 in Topology 1, (c) Gossip SGD for Device2 in Topology2, (d) Gossip Swap SGD for Device2 in Topology2

5. Conclusion

In this paper, we proposed a novel Gossip SGD method, named Gossip Swap SGD, that can resolve the non-convergence problem for decentralized distributed machine learning in heterogeneously dataset environments. We employed weights swapping (or exchanging) between devices that hold different dataset to update the weights. Our evaluation demonstrated the effectiveness of our method against the original Gossip SGD. In our future work, we will extend our method to cope with sparser WSN environments where weight swapping can be done less frequently.

Acknowledgment

This work is partially supported by JSPS KAKENHI JP20H04154.

References

[1] Chen, J. and Ran, X.: Deep Learning With Edge Computing: A Review, *Proceedings of the IEEE*, Vol. 107, No. 8, pp. 1655–1674 (2019).

[2] Chiu et al.: Semisupervised Distributed Learning With Non-IID Data for AIoT Service Platform, *IEEE Internet of Things Journal*, Vol. 7, No. 10, pp. 9266–9277 (2020).

[3] Jin, P. H. et al.: How to scale distributed deep learning?, *CoRR*, Vol. abs/1611.04581 (2016).

[4] McMahan, H. B. et al.: Federated Learning of Deep Networks using Model Averaging, *CoRR*, Vol. abs/1602.05629 (2016).